CS209A Final Project (2022 Spring)

Overview

We have learnt a lot of Java (and advanced Java) this semester. Hopefully this has been a delightful learning experience for you.

If you have trouble grasping a certain Java concept, or you've had a headache on a tricky bug that "mysteriously" crashes your program for no reason, don't worry, you are not alone. The same kind of frustration happens for practically every Java learner, from junior students to experienced professional developers. In fact, these "frustrations" and the corresponding solutions actually leave a trace on the Web, which gives us the opportunity to analyze and to reflect.

In this project, you'll become a data scientist to analyze the problems, solutions, opinions, and other development activities of the **Java programming language**. Possible topics include (**but not limited to**):

- Which Java features or concepts are mostly asked about?
- What are the most popular Java projects or applications in 2021?
- What are the most used Java techniques or framework for data visualization?
- What is the most challenging part for learning Java's web development?
- How do developers react when a new Java feature is firstly introduced or when a common Java feature becomes obsolete?
- For a specific Java project, how many bugs have been found and how many are fixed?
- .....

As a data scientist, you are responsible to find an interesting topic on Java (**use your imagination!**) and use your Java programming skills to collect, clean, analyze, and visualize related data. Finally, you are expected to derive useful insights and actionable knowledge from your analysis results.

Data Source

You could use any website as your data source (e.g., Stack Overflow, GitHub, CSDN, Zhihu, Baidu, etc.). You could use either web scraping or RESTful API to collect the data.

Nevertheless, we highly recommend you to explore Stack Overflow and GitHub, which are dedicated to host programming-related activities and discussions. You could use their official RESTful API for data collection.

- **Stack Overflow** is a Q&A site for programming questions and answers. Questions are classified mainly use tags. For example, a question on Java concurrency will have the tags `java` and `concurrency`. Check their official REST API documentation for more details.
- **GitHub** is a website for developers to store and manage their code. Developers could also use GitHub to track the releases, versions, issues, commits (code changes) and discussions of their projects. Check their official REST API documentation for more details.
- You may have to create an account for these data sources in order to use their full REST API service.
- Each data source has a daily quota for REST requests (e.g., 200 requests per day). Please carefully design and execute your requests.
- In case the connections to REST service are unstable, we also provide alternative URLs for accessing the service. Try use https://api.stackoverflow.arexh.top:3362/ for Stack Overflow REST service and https://api.github.arexh.top:3362/ for GitHub REST service if you are having trouble accessing the original services.

Techniques

- Backend functionalities, such as data collections and analysis, **must be implemented by Java**.
- Frontend functionalities, such as data visualization and interactive controls, could be implemented in any programming language (e.g., Java, JSP, HTML, JavaScript, CSS, etc.).
- Frontend visualization could be window-based or web-based (i.e., as a website opened in a browser).
- You could use any 3rd-party Java library or framework for your project.

Evaluation (Total points:25)

1. **Data collection, I/O and persistence (5 points)**: Data should be automatically collected from one or more data sources and stored in meaningful data structures. There should also be proper I/O methods for reading and writing data (e.g., writing the analysis results back to files or database).
2. **Data manipulation and analysis (5 points)**: Applying various algorithms and techniques to analyze different types of data (e.g., numerical data, textual data, etc.).

For instance, you could compute the descriptive statistics for numerical data, or applying NLP techniques to analyze textual data.

3. **Data visualization & user experience (5 points)**: You could use different types of charts to demonstrate the data analysis results. You could also add functionalities to support users' interaction with your application, e.g., users could search, sort, highlight, or export the analysis results.
4. **Topics, insights, and written report (5 points)**: You should provide a written report that describes the topics you selected for this project and the insights you obtain from the data analysis results. The written report should also introduce the architecture design of your project, as well as the important classes, fields, and methods.
5. **Presentation (5 points)**: Finally, your team should introduce your project to the class by delivering an oral presentation.

Teamwork

We encourage you to work in a team for this final project. The preferred team size is 2, while a team of 3 or a team of only 1 student is also allowed. However, teams of size 3 will get a 90% discount on their project scores, because the average workload for each student decreases. Teams of only 1 student will **NOT** get a bonus, because s/he doesn't have to make the communication efforts that are costly but crucial for a teamwork.

Sample data and visualization

We provide sample data and visualization as below. However, try **NOT** to directly reuse the exact data or visualization in your project. Use your imagination. We hope that each team could deliver a unique, creative, and insightful project.

- `resources_github_q_language_java.json` : This is the result obtained from executing the REST request: https://api.github.com/search/repositories?q=language:java
- `index.html` : This is a sample visualization for the sample data.

Submission

Please submit a zip file named **"StudentID-Name-Project.zip"** to Sakai. The submitted zip should include two parts:

1. **The project folder**, which includes all the source code and other relevant files necessary for running the project.
2. **A written report** (.pdf).